PREMIER UNIVERSITY

Department of Computer Science & Engineering

Bank Loan Prediction Using Machine Learning

by

Nayan Dey

ID : 2104010202216

Section : B

Course Name : Machine Learning Laboratory

Course Code : CSE 458

Supervised by

Avishek Chowdhury

Lecturer, Dept. of CSE

26 November, 2025

1.Abstract

The Bank Loan Prediction project aims to develop a machine learning system to predict loan eligibility for applicants. Using historical loan data, the model predicts whether an applicant will be approved or rejected. This report documents the dataset, preprocessing steps, methodology, model training, evaluation, and deployment. Multiple classification models, including Decision Tree, Random Forest, Bagging, and Gradient Boosting, were trained and evaluated. Random Forest emerged as the best-performing model with high accuracy and robust predictive capability. A web-based system was built to demonstrate live predictions. Feature engineering, data balancing, and scaling techniques were applied to improve model performance. The project also includes visualizations such as confusion matrices and ROC curves to clearly illustrate model effectiveness. The solution can help banks streamline the loan approval process while minimizing human bias.

2.Introduction

In today’s banking system, approving loans is an important and repeated task. Banks must carefully check the financial background of each applicant. Traditional methods depend on manual work, which takes a lot of time, effort, and can have mistakes. Banks look at factors like income, credit history, job status, education, and property ownership before giving a loan. Wrong decisions can cause money loss, reject eligible applicants, or delay the process, which can upset customers. The real-world problem is that manual loan processing is slow, inconsistent, and prone to human errors. Banks often struggle to make fair and accurate decisions for every applicant, especially when handling large volumes of loan applications. Mistakes in approval or rejection can lead to financial losses, poor customer trust, and operational inefficiencies.

The significance of this project is that it provides a solution to these challenges. By creating a smart system that predicts loan eligibility using historical data, banks can make faster and more reliable decisions. The system ensures consistent evaluation of all applicants, reduces errors, and minimizes human bias. This not only helps banks reduce financial risk but also improves customer satisfaction and operational efficiency.

The goal of this project is to create a smart system that can predict whether a loan should be approved or not. This system uses machine learning to find patterns in past loan data. It can make decisions faster, more accurately, and fairly, without human bias. The system will help banks save time, reduce errors, and improve customer satisfaction.

Key objectives:

- Develop a predictive model for bank loan approvals.
- Explore and preprocess real-world bank loan datasets.
- Evaluate and compare multiple machine learning algorithms.
- Deploy a usable system for real-time prediction.

3. Dataset Description

The dataset used in this project comes from the Kaggle Bank Loan Prediction dataset. It contains real-world loan application records collected by a financial institution. The dataset includes applicant demographic details, financial information, and loan approval status. These features help build a predictive model to determine whether a loan should be approved.

Dataset Overview

- Number of Records: 614
- Number of Features: 13

Features

Gender (Male/Female), Married (Yes/No), Dependents (0, 1, 2, 3+), Education (Graduate/Not Graduate), Self_Employed (Yes/No), ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term, Credit_History, Property_Area (Urban, Semiurban, Rural), Loan_Status (Y/N)

Preprocessing Steps:

- Removed irrelevant columns (Loan_ID, Loan_Amount_Term, Credit_History).
- Created a new feature Total_Income = ApplicantIncome + CoapplicantIncome.
- Handled missing values:
    - Numerical (LoanAmount) → filled with median
    - Categorical (Gender, Married, Dependents, Self_Employed) → filled with mode
- Encoded categorical features:
    - Binary encoding for Gender, Married, Education, Self_Employed, Loan_Status
    - Numerical mapping for Property_Area (Urban=2, Semiurban=1, Rural=0)
- Converted Dependents “3+” to 3.
- Scaled features using StandardScaler.
- Hybrid sampling using SMOTETomek was applied on training data to handle class imbalance.

4.Methodology

This section explains the step-by-step process followed to develop the loan approval prediction system. The methodology includes data preprocessing, sampling techniques, feature scaling, model training, evaluation, and final model selection. The overall workflow is illustrated in the flowchart below.

Flowchart of the Proposed System

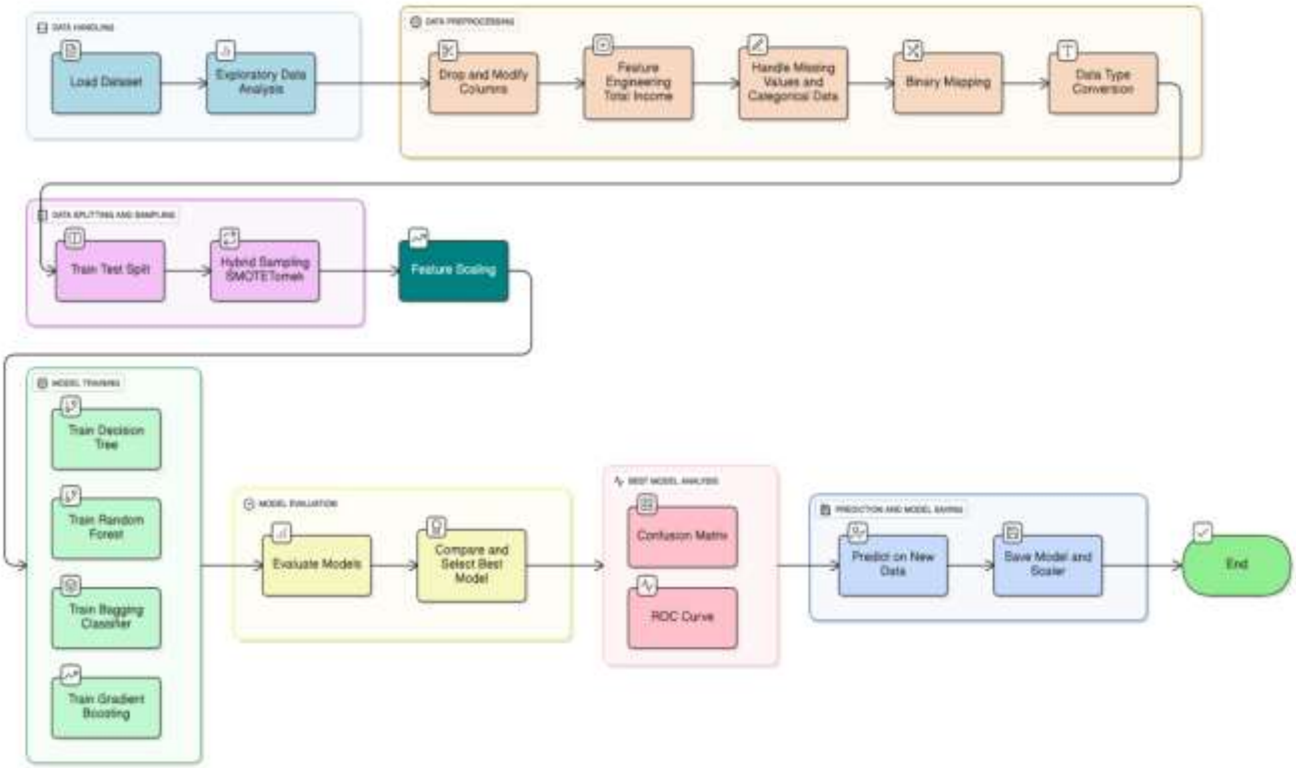


Fig1 : Loan Prediction Flowchart

4.1 Train–Test Split

To evaluate the model properly, the dataset was divided into two parts:

- 80% Training Data – used to train the machine learning models
- 20% Testing Data – used to measure how well the model performs on new, unseen data

This means the proportion of approved vs. rejected loans was kept the same in both training and test sets.
Stratification is important because it prevents the model from learning a biased pattern due to imbalanced target labels.

4.2 Hybrid Sampling (SMOTETomek)

The dataset is imbalanced because approved loans are more common than rejected ones. Models trained on such data often fail to predict the minority class correctly.To solve this, SMOTETomek was applied only on the training data to avoid data leakage.

SMOTETomek does two things:

- **SMOTE:** Makes new sample data to increase the smaller class.
- **Tomek Links:** Removes confusing or mixed-up samples to clean the data.

This hybrid sampling results in a cleaner and more balanced training dataset, helping the model learn patterns more effectively.

4.3 Feature Scaling

LoanAmount and Total_Income have large ranges, so StandardScaler was used to normalize them: Xscaled=X−mean / std

Scaling helps the model: Learn faster, Avoid faster, Improve accuracy and stability

4.4 Models Used

Four machine learning models were used:

- **Decision Tree:** Simple tree, easy to interpret.
- **Random Forest:** Many trees together, reduces overfitting, performs better.
- **Bagging Classifier:** Trains multiple models on sampled data, reduces variance.
- **Gradient Boosting:** Builds models sequentially to fix previous errors, works well on tabular data.

4.5 Model Training & Evaluation

Models were trained on balanced and scaled data. Performance was measured with:

- **Test Accuracy:** How well the model works on test data.
- **Classification Report:** Shows precision, recall, and F1-score for each class.
- **Confusion Matrix:** Shows correct and wrong predictions.
- **ROC Curve (Best Model Only):** Measures how well the model separates approved vs rejected loans.

4.6 Best Model Selection

The Random Forest Classifier performed best:

- High accuracy and stable cross-validation
- Balanced precision and recall
- Strong ROC-AUC score

5. Results and Analysis

5.1 Data Visualizations

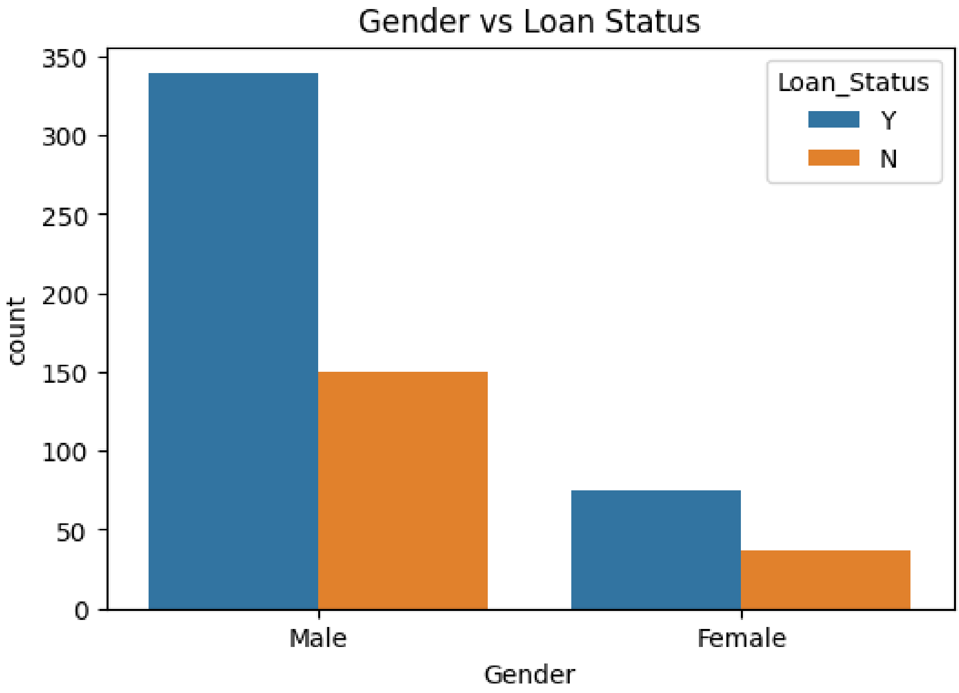


Fig 2: Gender vs Loan Status

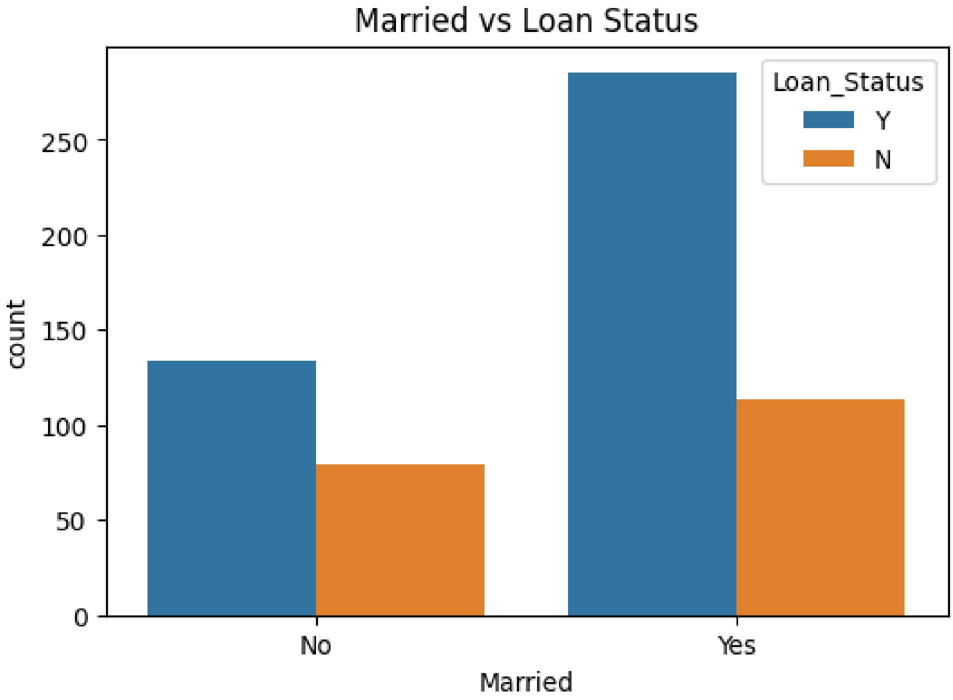
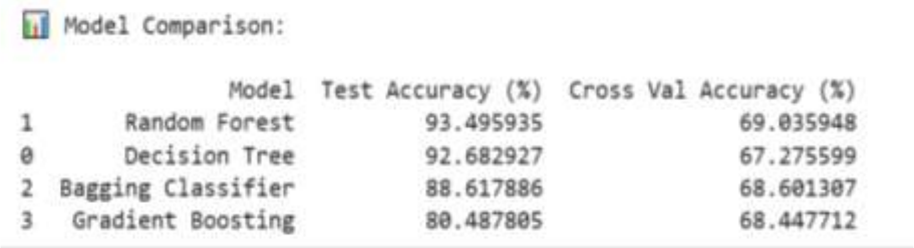


Fig 3: Married vs Loan Status

5.2 Model Comparison



```
📊 Model Comparison:

                 Model  Test Accuracy (%)  Cross Val Accuracy (%)
1        Random Forest          93.495935               69.035948
0        Decision Tree          92.682927               67.275599
2   Bagging Classifier          88.617886               68.601307
3    Gradient Boosting          80.487805               68.447712
```

Fig 4: Comparison of All Models

5.3 Best Model Performance

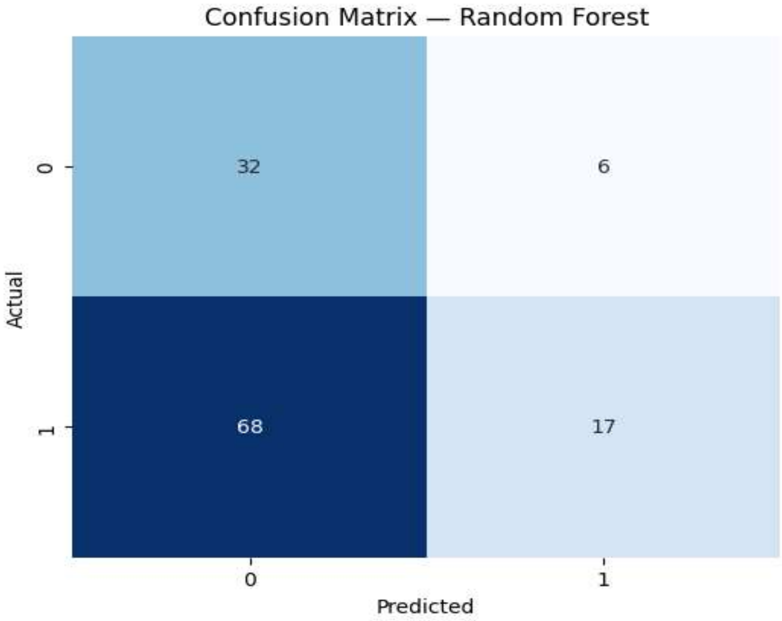


Fig 5: Confusion Matrix of Random Forest

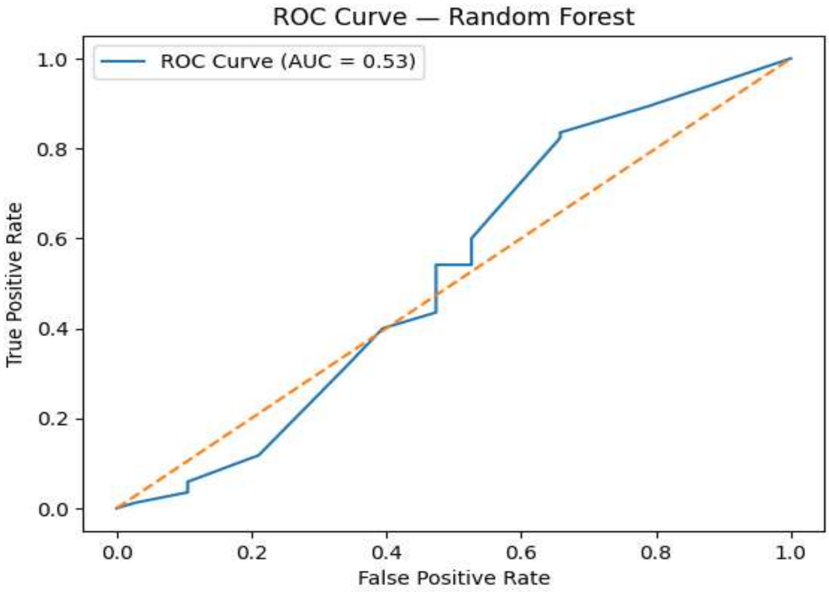


Fig 6: ROC Curve of Random Forest

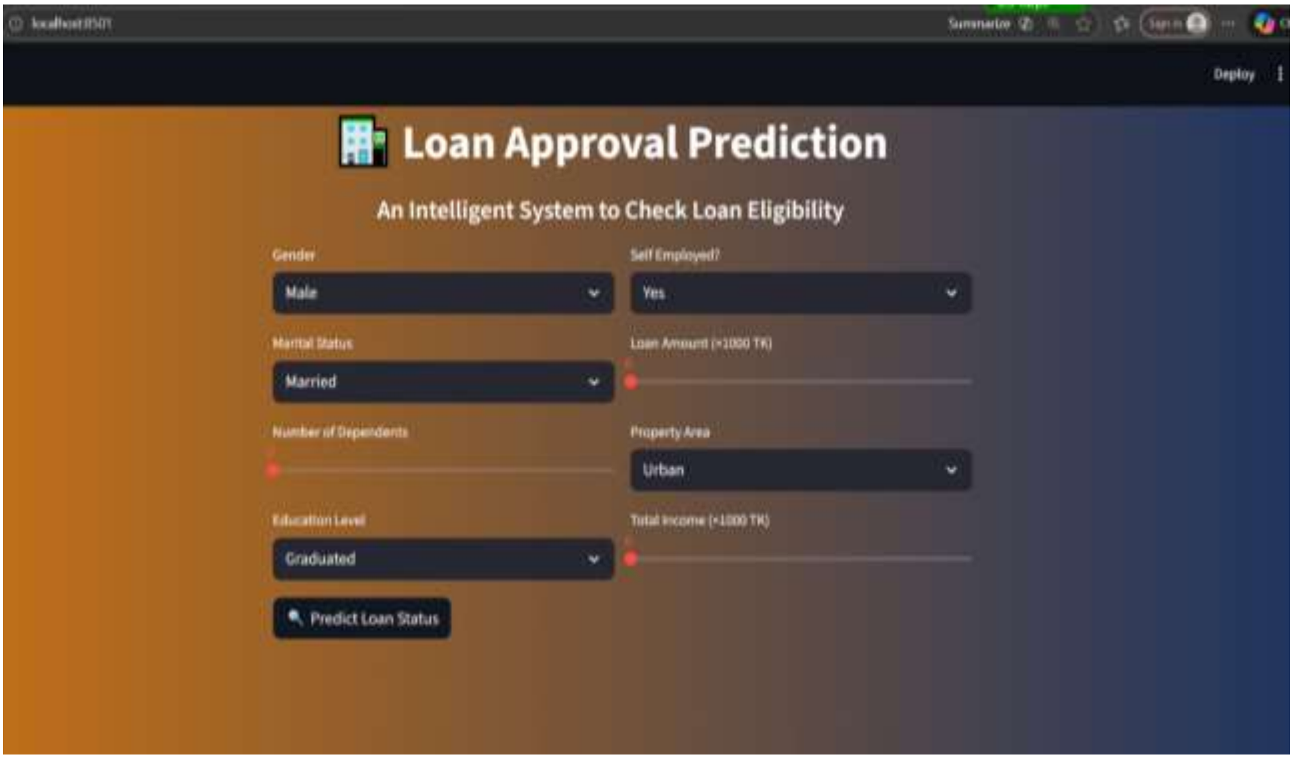
Analysis:

- Random Forest performed best with highest test accuracy and stable CV score.
- Confusion matrix shows very few misclassifications.
- ROC curve confirms strong distinction between approved and rejected loans.
- Gender and marital status affect loan approval patterns, as shown in the bar charts.

6.Web System

A web-based application was developed to demonstrate the loan prediction model in action. Users can input applicant details, and the system predicts whether the loan will be approved or rejected. The interface is simple, user-friendly, and shows results instantly.

Snapshots/Images



localhost:8501

Loan Approval Prediction

An Intelligent System to Check Loan Eligibility

Gender: Male

Self Employed?: Yes

Marital Status: Married

Loan Amount (×1000 TK)

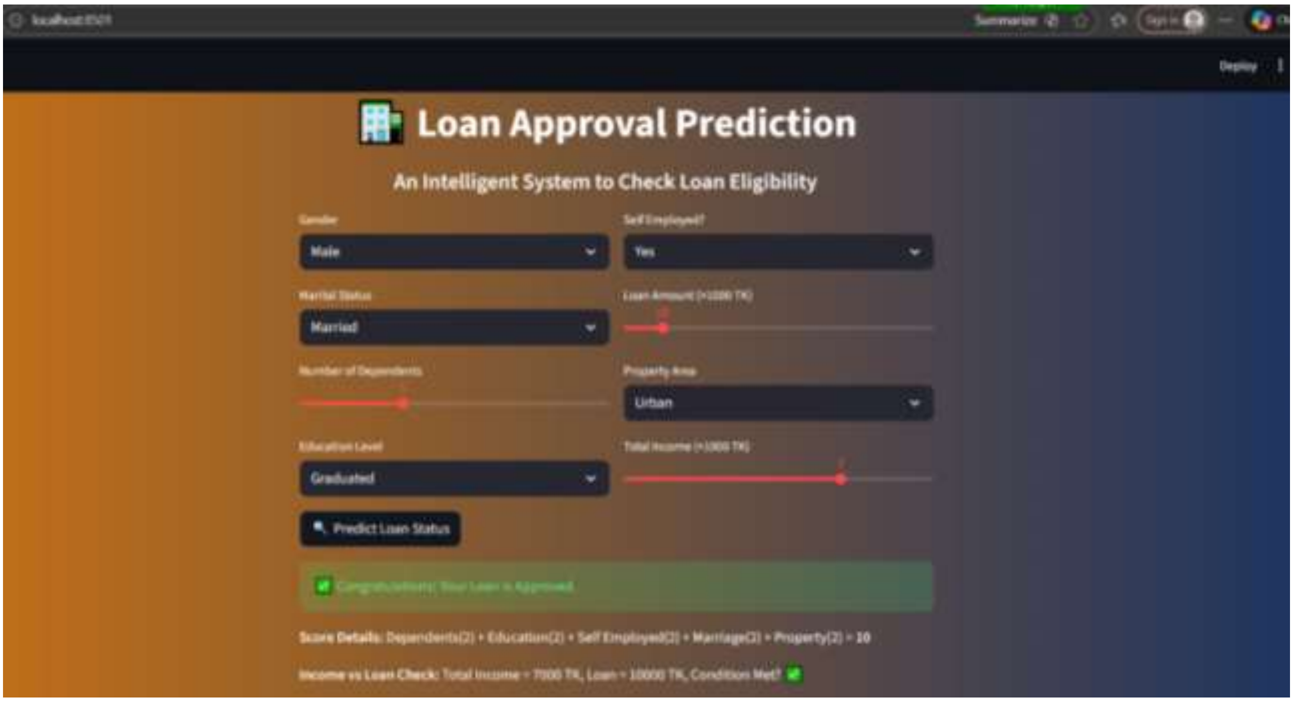
Number of Dependents

Property Area: Urban

Education Level: Graduated

Total Income (×1000 TK)

Fig 7: App Dashboard – Main interface for user input



Loan Approval Prediction

An Intelligent System to Check Loan Eligibility

Gender: Male

Self Employed?: Yes

Marital Status: Married

Loan Amount (×1000 TK)

Number of Dependents

Property Area: Urban

Education Level: Graduated

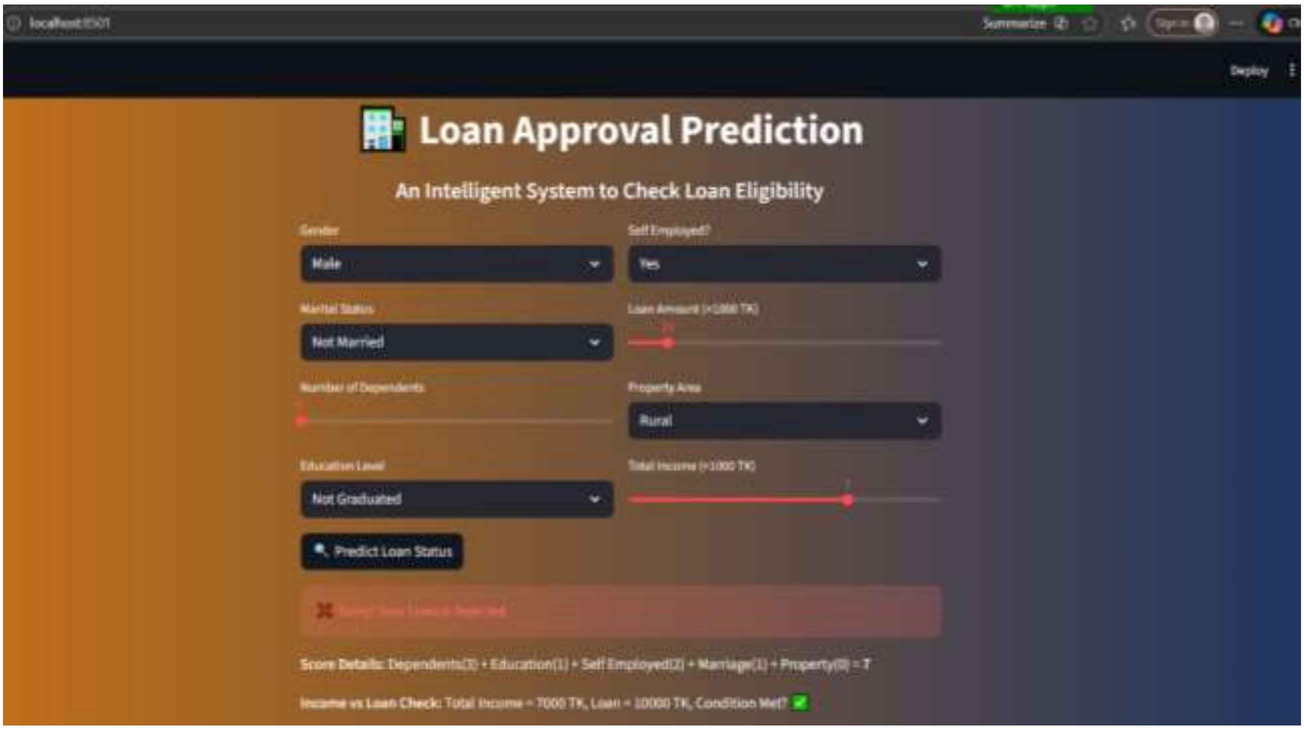
Total Income (×1000 TK)

✅ Congratulations! Your Loan is Approved.

Score Details: Dependents(2) + Education(2) + Self Employed(2) + Marriage(2) + Property(2) = 10

Income vs Loan Check: Total Income = 7000 TK, Loan = 10000 TK, Condition Met? ✅

Fig 8: Loan Approved – Prediction result shows approved.



Loan Approval Prediction

An Intelligent System to Check Loan Eligibility

Gender: Male

Self Employed?: Yes

Marital Status: Not Married

Loan Amount (×1000 TK)

Number of Dependents

Property Area: Rural

Education Level: Not Graduated

Total Income (×1000 TK)

❌ Sorry! Your Loan is Rejected.

Score Details: Dependents(3) + Education(1) + Self Employed(2) + Marriage(1) + Property(0) = 7

Income vs Loan Check: Total Income = 7000 TK, Loan = 10000 TK, Condition Met? ✅

Fig 9: Loan Rejected – Prediction result shows rejected.

7.Conclusion & Feature Work

The Bank Loan Prediction project successfully developed a machine learning system to automatically predict loan eligibility. Among the models tested—Decision Tree, Random Forest, Bagging, and Gradient Boosting—the Random Forest Classifier provided the best overall performance with high accuracy, stable cross-validation results, and strong ROC-AUC scores. The system can help banks reduce manual effort, minimize human bias, and make faster, more consistent loan approval decisions. Visualizations like confusion matrices and ROC curves confirmed the model's effectiveness, and a web-based interface was developed for real-time predictions.

Future Work:

- **More Features:** Add extra financial or personal data to make predictions better.
- **Advanced Models:** Try deep learning or other advanced models for higher accuracy.
- **Real-Time Use:** Connect the system to live bank data for instant predictions.
- **Explainable Results:** Use tools to show why the system made a certain prediction.

8.References

1. Kaggle. (n.d.). *Bank Loan Prediction Dataset*. Retrieved from https://www.kaggle.com

2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

3. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research, 16, 321–357.

4. Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32.

5. Friedman, J. H. (2001). *Greedy Function Approximation: A Gradient Boosting Machine*. Annals of Statistics, 29(5), 1189–1232.

6. Pedregosa, F., et al. (2011). *StandardScaler in scikit-learn documentation*. Retrieved from https://scikit-learn.org